Amendments to the Drawings:

The attached sheets of drawings include changes to Figures 1-5. These sheets, which include Figures 1-5, replace the original sheets including Figures 1-5.

Attachment: Replacement Sheets

REMARKS

Claims 1-22 and 26-33 are pending in this application. Claims 23-25 are canceled. Claims 27, 29 and 32 are amended.

The Examiner objected to Figures 1-5. Applicants have amended Figures 1-5 to indicate Figures 1-5 show the prior art.

On May 21, 2007, Applicants filed an Amendment in response to an Office Action dated February 20, 2007. Applicants subsequently noticed an error in the redlining of claims 1 and 2, with some text inadvertently appearing as underlined and stricken through in claims 1 and 2. Applicants left the Examiner a telephone message advising the Examiner of the error. The underlined and stricken through text was not intended to be included in claims 1 and 2, and is not included in the claims 1 and 2 as presented above. If the Examiner has concerns regarding the language of claims 1 and 2, Applicants respectfully request a telephone conference with the Examiner.

The Examiner rejected claims 12-22 and 27-32 under 35 U.S.C. Section 102(a) as anticipated by U.S. Patent No. 6,298,370 issued to Tang et al. The Examiner rejected claims 1-11, 26 and 33 under 35 U.S.C. Section 103(a) as obvious over U.S. Patent No. 6,298,370 issued to Tang et al., in view of U.S. Patent No. 6,799,266 issued to Stotzer. Applicants respectfully traverse the Examiner's rejections.

As an initial matter, Tang is an inappropriate primary reference. Tang is directed in general to a PC architecture. With regard to claims 1-14 and 26-28, Tang does not teach, suggest or motivate a first processor configured to modify at run time at least some of the compiled instruction words and convert them into modified instruction words executable on a second processor. Instead, Tang teaches compiling both x86 and DSP versions of granules and selecting one of the versions. Tang says nothing about modifying compiled instructions at runtime.

With regard to claims 15-18, 29 and 30, the Examiner's citations suggest the Examiner is relying on U.S. Patent No. 5,951,674 issued to Moreno and Tang, and Applicants will respond accordingly. Applicants respectfully request, to the extent the Examiner intended to

rely on Moreno or some combination of Moreno, Tang and Stotzer, that the Examiner withdraw the finality of the Office Action. In any event, with regard to claims 15-22 and 29-33, Tang is not an appropriate primary reference because the cited portions of Tang do not teach, suggest or motivate a processor in a plurality of processors directing instruction sets to processors in the plurality of processors based on workload data. Further, the Examiner has not provided any teaching, suggestion or motivation to combine Moreno and Tang in a manner so as to achieve the claimed invention.

Specifically, independent claim 1 recites, "retrieving a first set of instructions as compiled instruction words having a first length and executable on a first processor of said plurality; and modifying, during runtime of one of the programs, at least some of the instruction words of the first set of instructions by converting them into modified-instruction words executable on a second processor of said plurality." Independent claim 12 similarly recites, "means for converting instruction words of the first length compiled for execution on the first processor into modified instruction words of the second length executable on the second processor."

The Examiner points to column 3, lines 25-55 and column 20, lines 20-30 of Tang as allegedly teaching "modifying, during runtime of one of the programs, at least some of the instruction words of the first set of instructions by converting them into modified-instruction words executable on a second processor of said plurality." Column 3, lines 25-55 of Tang contains the summary of the invention, and recites:

> Generally, and in one form of the present invention, a process of operating a computer system is provided. The computer system has a storage holding an operating system and an application program, a first processor having an instruction set, and a second processor having a different instruction set. The process includes first step of 1) running at least some of the operating system on the first processor so that the first processor sets up for at least part of the application program at run time at least one second processor object. A second step concurrently runs the second processor to access the second processor object and thereby determine operations for the second processor to access second

> processor instructions for that part of the application program and data to be processed according to the second processor instructions, and runs the second processor to process the data according to the second processor instructions.
>
> Tang, Column 3, lines 34-50.

Tang goes on to explain that an application program is loaded on a CPU and that during execution part of the application program (so-called granules) might be executed on a digital signal processor:

> DirectDSP extends DirectX to intelligently distribute processing MIPs between the host CPU/MMX and the VSP(s) by parsing tasks into sub-tasks (granules) which then are run by either the host or VSP(s) in a dynamic and balanced fashion. Both host and VSP application granules are called by DirectDSP/DirectDSP HAL using multitasking and multithreaded Windows OS, COM-based (Component Object Model) DirectX and ActiveX as well as the host CPU/MMX and PC core logic. Direct DSP runs on top of the DirectDSP HAL or the DirectDSP WDM stack.

Tang, Col. 23, lines 51-60. In computer science "parsing" (more formally syntactic analysis) is the process of analyzing a sequence of tokens to determine its grammatical structure with respect to a given formal grammar. A parser is the component of a compiler that carries out this task.

The second section of Tang upon which the Examiner relies, Column 20, lines 20-30, recites:

> A software decoder, for example, has lower priority than that of a hardware event. The VSP by means of hardware interrupts can naturally preempt a host-based program and work to advantage in the Windows OS scheduler environment. The VSP briefly interrupts the host to raise its priority with the Windows OS scheduler. If the host were to lock out interrupts, it would simply become a single-tasking system, therefore the host should not do so. Thus, VSP is a "very good citizen" for the Windows OS.
>
> Software tasks are each largely broken into fine granules that are easily ***modified and compiled*** not only on an x86 compiler but also a DSP compiler."

Tang, Column 20, lines 20-30 (emphasis added). Thus, compiled instructions are not modified at runtime. Instead, software granules are modified and compiled for both architectures. *See also*, Column 27, lines 10-16 ("For example, such host granules written to perform USP sub-tasks as function are recompiled to run on the VSP as application granules"), and Figure 21.

In Tang, an application granule that has to be able to run on both the CPU and the DSP is compiled twice, once for each architecture. Tang does not teach, suggest or motivate ***modifying*** instruction words at run time, but instead ***selecting*** either x86 or DSP versions of compiled instruction granules. Accordingly, Tang, alone or in combination with Stotzer, does not render claims 1-11 and 26 obvious.

In addition, neither Tang nor Stotzer teach, suggest or motivate splitting said instruction words into modified-instruction words, as recited in claim 1. Further, one of skill in the art would not be motivated to combine Tang and Stotzer as suggested by the Examiner. Claim 1 recites, "entering in the modified-instruction words no-operation instructions." The Examiner relies on Stotzer as the motivation for inserting NOP instructions. Tang uses C-source code, which is compiled for both the CPU and the DSP as discussed above. Introducing NOP instructions in Tang during compilation or execution would provide no technical advantage, and would not achieve the claimed invention. Thus, claim 1, and claims 2-11 and 26 which depend from claim 1, are not rendered obvious over Tang, alone or in combination with Stotzer, for these additional reasons.

Similarly, claims 12-14, 27 and 28 are not anticipated or rendered obvious by Tang, alone or in combination with Stotzer, because the combination of Tang and Stotzer does not teach suggest or motivate "means for converting instruction words of the first length compiled for execution on the first processor into modified instruction words of the second length executable on the second processor," as recited in claim 12. Tang teaches recompiling granules, and does not teach, suggest or motivate modifying compiled instruction words. Claims 13, 14, 27 and 28 depend from claim 12 and are allowable at least by virtue of their dependencies.

Claim 13 further recites, "said processors are all of the Very Long Instruction Word (VLIW) type." The Examiner points to Figure 21 of Tang and the accompanying

description. The description of Figure 21 indicates that the DSP 2110 is coupled to CPU/MMX 315. Thus, claim 13 is not anticipated or rendered obvious by Tang, alone or in combination with Stotzer for the additional reason that Tang, alone or in combination with Stotzer, does not teach, suggest or motivate all of the processors in the plurality being of a VLIW type.

In addition, claim 27 recites, "said means for converting instruction words of the first length compiled for execution on the first processor into modified instruction words of the second length executable on the second processor is configured to convert the instruction words during runtime." As discussed above with regard to claim 1, Tang, alone or in combination with Stotzer, does not teach, suggest or motivate modifying instruction words at runtime.

Independent claim 15 recites, "a plurality of processors coupled for receiving compiled instruction sets; a first processor of the plurality coupled to each of the other processors within said plurality, said first processor receiving from the other processors data representative of the workload of each of said other processors; an output signal from said first processor to said instruction set stream, said output signal controlling the instructions, which are sent to each of said processors based on the results of the workload measurement of said processors."

The Examiner points to Tang, Figure 6, element 606, Figure 16, element 706, and the related text, to Column 3, lines 35-50, and to Figure 21 and the related text. There is no element 606 in Figure 6, and no element 706 in Figure 16 of Tang. Moreover, the cited portions of Tang do not teach, suggest or motivate a first processor receiving from each of the other processors data representative of workload and an output signal from the first processor to the instruction set stream. Based on the citations provided by the Examiner and a prior Office Action mailed on February 20, 2007, the Examiner appears to be relying on Moreno rather than Tang. Moreno, alone or in combination with Tang, does not teach, suggest or motivate receiving a compiled instruction sets, a first processor receiving data representative of the workload and generating an output signal controlling the instructions. Accordingly, claims 15-18 and 29-30 are not anticipated by Tang, or rendered obvious by Tang in view of Moreno. Claim 29 is allowable for the additional reason that Tang does not teach suggest or motivate each of the processors being of a VLIW type, as discussed above with regard to claim 13.

Independent claim 19 recites, "receiving a plurality of executable instruction sets on a bus line connected to said processors; receiving workload data at a first processor of said plurality of processors, said workload data being representative of workload of each of the processors of said plurality; comparing the workload of each of the processors; and sending a signal from said first processor based on the data representative of the workload of each of the processors of said plurality to the bus line for modifying the number of executable instruction sets sent to each processor based on their respective workloads." The Examiner points to Figures 1, 21 and 124 of Tang and the accompanying text. The cited portions of Tang do not teach, suggest or motivate receiving a plurality of executable instruction sets, receiving workload data representative of the workload of each of the processors at a first processor, comparing the workload of each of the processors, and sending a signal from the first processor for modifying the number of executable instructions sent to each processor based on their respective workloads. The Examiner points to Figure 124 as teaching comparing the workload of each of the processors. Figure 124 discusses altering the speed of a DSP based on a lookup table of the speed required for a DSP-enabled application. There is no discussion of comparing the workload of processors in a plurality of processors. Claims 20-22 and 31-33 depend from claim 19 and are allowable at least by virtue of their dependencies. Claim 32 is allowable for the additional reason that Tang does not teach suggest or motivate each of the processors being of a VLIW type, as discussed above with regard to claim 13.

The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited. If the Examiner disagrees or has any concerns with formalities, Applicants respectfully request a telephone conference with the Examiner.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC

Timothy L. Boller
Registration No. 47,435

TLB:jms

Enclosure:
3 Sheets of Replacement Drawings (Figures 1-5)

701 Fifth Avenue, Suite 5400
Seattle, Washington 98104
Phone: (206) 622-4900
Fax: (206) 682-6031

1019372_1.DOC